

Document No. AUS000181US1

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Snow**

Serial No. **09/552,861**

Filed: **April 20, 2000**

For: **Application Development Server
and a Mechanism for Providing
Different Views into the Same
Constructs Within a Strongly
Encapsulated Environment**

§
§
§
§
§
§
§

Group Art Unit: **2122**

Examiner: **Rutten, James D.**

**Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450**

Certificate of Mailing Under 37 C.F.R. § 1.8(a)

I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on April 1, 2005.

By:

Amelia C. Turner
Amelia C. Turner

04/05/2005 FMETEKI1 00000009 090447 09552861

01 FC:1402 500.00 DA

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal, filed in this case on February 1, 2005.

The fees required under § 41.20(B)(2), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation, as reflected in the Assignment recorded on April 20, 2000, at Reel 010762, Frame 0015.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1, 2, 5, 6, 9, 11-13, and 25-35.

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: 3-4, 7-8, 10, and 14-24.
2. Claims withdrawn from consideration but not canceled: None.
3. Claims pending: 1, 2, 5, 6, 9, 11-13, and 25-35.
4. Claims allowed: None.
5. Claims rejected: 1, 2, 5, 6, 9, 11-13, and 25-35.
6. Claims objected to: None.

C. CLAIMS ON APPEAL

The claims on appeal are: 1, 2, 5, 6, 9, 11-13, and 25-35.

STATUS OF AMENDMENTS

An amendment after final is filed herewith.

SUMMARY OF CLAIMED SUBJECT MATTER

Applicant's claims 1, 25, and 30 describe a method, server, and product for providing automated software development to a client. The following description refers to claims 1, 25, and 30. A storage unit is described for storing a plurality of working definitions for a plurality of computing environments. See Figures 4A and 7A. An interface is described for receiving requests for one of the computing environments from a client and transferring the requested computing environment to the client over a communications link.

Each one of the computing environments includes a set of first working definitions that defines a plurality of applications (for example, see Figure 4A and specification page 14, lines 17-30) and a second working definition that defines the computing environment (for example, see Figure 7A and specification page 26, lines 1-15). The set of first working definitions and the second working definition are included in the plurality of working definitions.

The second working definition defines the requested computing environment. Further, the second working definition is included in the computing environment itself.

Each one of the first working definitions includes platform independent definitions that define characteristics, of one of the plurality of applications that are necessary to construct a valid runtime image of the application. The characteristics include state, settings, and structures that are required to build the runtime image of the application. The second working definition includes platform independent definitions that define characteristics of the computing environment that are necessary to construct a valid runtime image of the computing environment. The characteristics include state, settings, and structures required to build the runtime image of the computing environment. For example, see Figure 4A and specification page 14, lines 17-30, and Figure 7A and specification page 26, lines 1-15.

According to dependent claims 5, 26, and 31, each one of the first working definitions includes source code for one of the applications defined by the working definition.

According to dependent claims 6, 27, and 32, encrypted links are sent to the client along with the requested computing environment where the encrypted links provide support for development, testing, beta testing, and deployment of the plurality of applications.

According to dependent claims 11, 28, and 33, the interface receives an update to one of the first working definitions. That working definition is then modified to reflect the update.

According to dependent claims 12, 29, and 34, a change occurs in the data processing system. In response to a determination that the change affects one of the working definitions, that working definition is modified to reflect the change. The runtime image of the application defined by that working definition is updated.

According to dependent claims 13 and 35, the change is an event.

According to dependent claim 2, the communications link is the Internet.

According to dependent claim 9, the communications link is an intranet.

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

A. GROUND OF REJECTION 1 (Claims 1, 2, 5, 9, 11-13, 25, 26, 28-31, and 33-35)

Claims 1, 2, 5, 9, 11-13, 25, 26, 28-31, and 33-35 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,145,119 issued to *House* in view of U.S. Patent 5,361,360 issued to *Ishigami* and further in view of U.S. Patent 6,083,276 issued to *Davidson*.

B. GROUND OF REJECTION 2 (Claims 6, 27, and 32)

Claims 6, 27, and 32 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,145,119 issued to *House* in view of U.S. Patent 5,361,360 issued to *Ishigami* and U.S. Patent 6,083,276 issued to *Davidson* and further in view of U.S. Patent 5,423,042 issued to *Jalili*.

ARGUMENT

The Examiner rejected claims 31-35 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. Specifically, the Examiner noted that claim 31 is described as depending on claim 31. A Response to Final Office Action is filed herewith to amend claim 31 so that it depends on claim 30. Applicant believes this Response to Final Office Action will be entered as it removes issues for appeal. Therefore, the claims appended to this appeal brief reflect the amendment to claim 31. Because Applicant believes that the Response to Final Office Action filed herewith will result in the withdrawal of this rejection, this rejection will not be addressed further in this appeal brief.

A. GROUND OF REJECTION 1 (Claims 1, 2, 5, 9, 11-13, 25, 26, 28-31, and 33-35)

Claims 1, 2, 5, 9, 11-13, 25, 26, 28-31, and 33-35 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,145,119 issued to *House* in view of U.S. Patent 5,361,360 issued to *Ishigami* and further in view of U.S. Patent 6,083,276 issued to *Davidson*. This position is not well founded.

Applicant claims a plurality of computing environment. Each computing environment includes a set of first working definitions that define a plurality of applications. Each computing environment also includes a second working definition that defines its computing environment.

Each first working definition includes platform independent definitions that define the characteristics of one of the applications. These are characteristics that are necessary to construct a valid runtime image of the application, including state, settings, and structures that are required in order to build the runtime image of the one application.

The second working definition includes platform independent definitions that define the characteristics of its computing environment. These are characteristics that are necessary to construct a valid runtime image of the computing environment, including state, settings, and structures that are required in order to build the runtime image. The second working definition is included in the computing environment itself.

The Examiner relies on *House* as the primary reference and believes it teaches a plurality of computing environments and a computing environment that includes a set of first working definitions that define a plurality of applications. The Examiner relies on *Davidson* to teach platform independent definitions, and *Ishigami* to teach a second working definition that defines a computing environment. Applicant disagrees with the Examiner and believes that none of the reference teaches a plurality of computing environments or a computing environment that includes a set of first working definitions that define a plurality of applications.

House describes providing a project environment that supports development of internet applications. *House* describes a data structure, i.e. an APP file, for storing all aspects of a project. The data structure is for encapsulating a software project with a project application that is defined by executable programming logic, and includes a project environment for developing the application. Each data structure therefore defines one software project. Each data structure includes executable logic for one application. Each data structure also includes descriptive data that describes the project environment for developing the one application.

Applicant claims a plurality of computing environments. The Examiner states that a “computing environment” as claimed by Applicant is analogous to the “project environment” taught by *House*. The Examiner also states that “review of element 454 in FIG. 4 of *House* shows that one project environment can have a plurality of applications, or ‘APP files’.” See the Examiner’s Final Office Action page 4, paragraph 5. Thus, according to the Examiner’s argument, one computing environment, i.e. the project environment, can have multiple applications because it can include multiple APP files. In this case the Examiner appears to be arguing that each APP file is an application and the overall environment depicted by Figure 4 is the computing environment.

However, elsewhere in the Examiner’s action, the Examiner appears to be arguing that just one of the APP files is a computing environment. See the Examiner’s final action page 7, last paragraph through page 9, first paragraph.

Assuming for the sake of argument that Applicant’s computing environment is analogous to *House*’s “project environment” and that each APP file is an application, *House* does not teach these features of Applicant’s claims. Applicant claims each computing environment including a set of first working definitions that define a plurality of applications and a second working

definition that defines its computing environment. If each APP file is just an application, there is no second working definition described by *House* at all.

Further, *House* does not describe a plurality of computing environments if the computing environment of Applicant's claim is understood to mean the development computer of Figure 4. *House* does not teach a plurality of development computers.

This scenario where Applicant's computing environment is believed to be analogous to *House*'s "project environment" and each APP file is an application is not consistent with the remainder of the Examiner's argument. The Examiner states in paragraph 19, starting on page 7, that *House* discloses a plurality of computing environments by teaching an APP file 454. Applicant understands this statement to mean the Examiner believes an APP file 454 is analogous to a "computing environment", and not to a single application.

The scenario where a "computing environment" is analogous to *House*'s "project environment" and each APP file is an application is not consistent with *House*. *House* states that each APP file contains all of the data and structures that represent a software project. Column 6, lines 1-7. An APP file is not described by *House* as being merely an application. An APP file is the data structure referred to by *House* that comprises a first section which comprises executable logic to load and execute one application, and a second section that includes other data, including data required to restore the project environment for developing the application.

House teaches a data structure, i.e. an APP file, that includes two parts, a runtime section and an "other data" section. The runtime section can be used to run one application. The second section includes other data including data required to restore a project environment for developing the application. *House* teaches a single data structure being used to encapsulate all elements that are necessary to build one project environment. Therefore, there is one data structure to represent one project environment. Thus, an APP file is not merely an application, it is used to build one project environment.

Now assuming for the sake of argument that Applicant's computing environment is analogous to an APP file, *House* does not teach these features of Applicant's claims. Applicant claims the computing environment including a first set of working definitions that define a plurality of applications. One APP file of *House*, however, includes a first section of executable logic for executing only one application.

House does not teach a computing environment that includes a set of first working definitions that define a plurality of applications. Applicant claims a computing environment that includes a set of first working definitions that define a plurality of applications. According to *House*, a first section of the data structure, i.e. one APP file, includes the executable program logic while a second section of the structure includes all other data that is needed. An APP file is constructed that includes all of the data and structures that represent a software project including project related data such as information on the application for the server, stored procedures or user defined functions, source code, event logic controls, forms, and relationships among all of the things. From the single APP file 454, a project and all of its states can be recovered.

The Examiner states that *House* teaches the set of first working definitions by teaching the runtime section 550 of APP file 454, and teaches a second working definition by teaching the “other data” section 552 of APP file 454. The APP file 454 includes a runtime section 550 and an “other data” section 552. The runtime section contains all of the data that is needed at runtime to run one application, not a plurality of applications. The “other data” section includes all other project related data. When executing an application, the “other data” section can be skipped.

Each APP file of *House* includes executable logic for a single application and not a plurality of applications. Therefore, if each APP file is believed to be a computing environment, *House* does not teach a computing environment that includes a first set of working definitions that define a plurality of applications because each APP file includes executable logic for just one application.

According to Applicant’s claims, each one of the computing environments includes a second working definition that defines said each one of the computing environments. This is the computing environment that includes the second working definition. *House* does not teach a computing environment that includes a second working definition that defines its computing environment that includes the set of first working definitions that define a plurality of applications and the second working definition. Each APP file encapsulates an application and a project environment for developing the application. The runtime section includes the application and the “other data” section includes data required to restore the project environment.

The Examiner appears to believe the APP file is analogous to a computing environment. Therefore, for *House* to teach the feature of the second working definition defining its computing environment, the “other data” section of the APP file must define the APP file. However, this is not the case. This is not what is taught by *House*. *House* teaches its other data section as including data required to restore the project environment, and not the APP file itself.

House does not teach a second working definition that defines characteristics that are necessary to construct a valid runtime image of its computing environment. According to Applicant’s claims, both the first working definitions and the second working definition include definitions that are necessary in order to construct a valid runtime image. *House* does not teach a set of first working definitions and a second working definition that all include definitions that are necessary to construct a valid runtime image.

Data objects that comprise executable programming logic are stored in the runtime section 550. The remaining data that describes a development environment is stored in the “other data” section 552. In this manner, the runtime data is kept separate (kept in runtime section 550) from the descriptive, non-runtime, data (kept in “other data” section 552). When executing an application, reading in the “other data” section 552 is skipped. Therefore, it is clear that the “other data” section does not include runtime data.

House teaches away from the “other data” section including definitions that are necessary to construct a runtime image. *House* teaches the runtime data being included in the runtime section 550 and not in the “other data” section 552 that the Examiner states teaches Applicant’s “second working definition”.

The Examiner states that *House* does not teach platform independent definitions, or a second working definition that defines said each one of said plurality of computing environments. The Examiner relies on *Davidson*, column 6, lines 45-48, to teach the feature of platform independent definitions. However, the combination of *House* and *Davidson* does not teach a second working definition that is included in a computing environment that includes a set of first working definitions that define a plurality of applications where the working definitions include platform independent definitions.

The Examiner relies on *Ishigami*, column 2, lines 35-47, to teach a second working definition that defines a computing environment. *Ishigami* teaches storing definition information

of a programming environment including a set of names of tools and names of data that are necessary for developing software. According to a first aspect of *Ishigami*, tools and data are transmitted and installed. According to a second aspect of *Ishigami*, when the definition information is changed, tools and data that are already installed are updated.

In *Ishigami*, definitions may be received. These definitions define tools and data. Some of these tools and data may have already been stored. Thus, these definitions may include updated version of some of the stored tools and data. *Ishigami* does not teach requesting a computing environment that includes a first set of working definitions that define characteristics of a plurality of applications and a second working definition that defines characteristics of the requested computing environment. The section of the reference referred to by the Examiner describes definition information being received that includes updates of tools and data that are already stored. This definition information is not analogous to the second working definition of Applicant's claims.

Applicant describes the second working definition defining characteristics that are used to construct a runtime image of the requested computing environment. Further, the second working definition is included in the computing environment itself.

None of the references describes, teaches, or suggests a computing environment that includes a first set of working definitions that define a plurality of applications, and a second working definition that defines the computing environment itself where the first set of working definitions defines characteristics that are necessary to construct a valid runtime image of the applications and the second working definition defines characteristics that are necessary to construct a valid runtime image of the computing environment. Therefore, this combination of references does not render Applicant's claims unpatentable.

B. GROUND OF REJECTION 2 (Claims 6, 27, and 32)

Claims 6, 27, and 32 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,145,119 issued to *House* in view of U.S. Patent 5,361,360 issued to *Ishigami* and U.S. Patent 6,083,276 issued to *Davidson* and further in view of U.S. Patent 5,423,042 issued to *Jalili*. This position is not well founded.

These claims describe encrypted links that are sent to the client along with the requested computing environment where the encrypted links provide support for development, testing, beta testing, and deployment of the plurality of applications.


Jalili teaches remote procedure execution. The Examiner states that *Jalili* teaches a system of communication using encrypted links. Although *Jalili* does teach an encrypted identifier, *Jalili* does not teach the features of Applicant's claims that are missing from the other references. Applicant's claims do not merely describe encrypted links. These claims describe the encrypted links providing support for development, testing, beta testing, and deployment of the plurality of applications. These applications are defined by the set of first working definitions included in the requested computing environment.

The references, neither singly nor in combination, describe, teach or suggest encrypted links that provide support for development, testing, beta testing, and deployment of these plurality of applications in combination with the other features of Applicant's claims. Therefore, this combination of references does not render Applicant's claims unpatentable.

C. CONCLUSION

The combination of the cited references does not describe, teach, or suggest a computing environment that includes a first set of working definitions that define a plurality of applications, and a second working definition that defines the computing environment itself where the first set of working definitions defines characteristics that are necessary to construct a valid runtime image of the applications and the second working definition defines characteristics that are necessary to construct a valid runtime image of the computing environment. The combination of references also does not describe, teach, or suggest a computing environment that includes a first set of working definitions that define a plurality of applications, and a second working definition that defines the computing environment itself where the first set of working definitions defines characteristics that are necessary to construct a valid runtime image of the applications and the second working definition defines characteristics that are necessary to construct a valid runtime image of the computing environment in combination with encrypted links that are sent to the client along with the requested computing environment where the encrypted links provide support for development, testing, beta testing, and deployment of the plurality of applications.

Therefore, Applicant's claims are believed to be patentable over the cited prior art.



Lisa L.B. Yociss
Reg. No. 36,975
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 385-8777

CLAIMS APPENDIX

The text of the claims involved in the appeal reads:

1. An application development server, comprising:

a storage unit for storing a plurality of working definitions for a plurality of computing environments;

an interface for receiving requests for one of the plurality of computing environments from and transferring the requested one of the plurality of computing environments to a client over a communications line;

each one of said plurality of computing environments including a set of first working definitions that define a plurality of applications and a second working definition that defines said each one of said plurality of computing environments, said set of first working definitions and said second working definition being included in said plurality of working definitions;

each one of said set of first working definitions including platform independent definitions that define characteristics of one of said plurality of applications that are necessary to construct a valid runtime image of said one of said plurality of applications, said characteristics including state, settings, and structures required to build said runtime image of said one of said plurality of applications; and

said second working definition including platform independent definitions that define characteristics of said each one of said plurality of computing environments that are necessary to construct a valid runtime image of said each one of said plurality of computing environments, said characteristics including state, settings, and structures required to build said runtime image of said each one of said plurality of computing environments.

2. The application development server as recited in claim 1, wherein the communication line is the Internet.

5. The application development server as recited in claim 1, wherein each one of said set of first working definitions includes source code for one of said plurality of applications defined by said one of said set of first working definitions.

6. The application development server as recited in claim 1, wherein encrypted links are sent to the client along with the requested one of the plurality of computing environments, the encrypted links providing support for development, testing, beta testing, and deployment of the plurality of applications.

9. The application development server as recited in claim 1, wherein the communications line is an intranet.

11. The application development server as recited in claim 1, further comprising:
said interface receiving an update to one of said set of first working definitions; and
said one of said set of first working definitions being modified to reflect said update.

12. The application development server as recited in claim 1, further comprising:
a change occurring within a data processing system;

responsive to a determination that the change affects one of said set of first working definitions, the one of said set of first working definitions being modified to reflect the change; and

the runtime image of one of said plurality of applications defined by said one of said set of first working definitions being updated.

13. The application development server as recited in claim 12, wherein the change is an event.

25. A computer program product, comprising:

instruction means for storing a plurality of working definitions for a plurality of computing environments;

instruction means for receiving requests for one of the plurality of computing environments from and transferring the requested one of the plurality of computing environments to a client over a communications line;

each one of said plurality of computing environments including a set of first working definitions that define a plurality of applications and a second working definition that defines said each one of said plurality of computing environments, said set of first working definitions and said second working definition being included in said plurality of working definitions;

each one of said set of first working definitions including platform independent definitions that define characteristics of one of said plurality of applications that are necessary to construct a valid runtime image of said one of said plurality of applications, said characteristics

including state, settings, and structures required to build said runtime image of said one of said plurality of applications; and

said second working definition including platform independent definitions that define characteristics of said each one of said plurality of computing environments that are necessary to construct a valid runtime image of said each one of said plurality of computing environments, said characteristics including state, settings, and structures required to build said runtime image of said each one of said plurality of computing environments.

26. The product as recited in claim 25, wherein each one of said set of first working definitions includes source code for one of said plurality of applications defined by said one of said set of first working definitions.

27. The product as recited in claim 25, further comprising instruction means for sending encrypted links to the client along with the requested one of the plurality of computing environments, the encrypted links providing support for development, testing, beta testing, and deployment of the plurality of applications.

28. The product as recited in claim 25, further comprising:

instruction means for receiving an update to one of said set of first working definitions;
and

instruction means for modifying said one of said set of first working definitions to reflect said update.

29. The product as recited in claim 25, further comprising:

instruction means for detecting a change within a data processing system;

responsive to a determination that the change affects one of said set of first working definitions, instruction means for modifying the one of said set of first working definitions to reflect the change; and

instruction means for updating the runtime image of one of said plurality of applications defined by said one of said set of first working definitions.

30. A method for providing automated software development to a client, said method comprising:

storing a plurality of working definitions for a plurality of computing environments;

receiving requests for one of the plurality of computing environments from and transferring the requested one of the plurality of computing environments to a client over a communications line;

each one of said plurality of computing environments including a set of first working definitions that define a plurality of applications and a second working definition that defines said each one of said plurality of computing environments, said set of first working definitions and said second working definition being included in said plurality of working definitions;

each one of said set of first working definitions including platform independent definitions that define characteristics of one of said plurality of applications that are necessary to construct a valid runtime image of said one of said plurality of applications, said characteristics including state, settings, and structures required to build said runtime image of said one of said plurality of applications; and

said second working definition including platform independent definitions that define characteristics of said each one of said plurality of computing environments that are necessary to construct a valid runtime image of said each one of said plurality of computing environments, said characteristics including state, settings, and structures required to build said runtime image of said each one of said plurality of computing environments.

31. The method as recited in claim 30, wherein each one of said set of first working definitions includes source code for one of said plurality of applications defined by said one of said set of first working definitions.

32. The method as recited in claim 31, further comprising sending encrypted links to the client along with the requested one of the plurality of computing environments, the encrypted links providing support for development, testing, beta testing, and deployment of the plurality of applications.

33. The method as recited in claim 31, further comprising:
receiving an update to one of said set of first working definitions; and
modifying said one of said set of first working definitions to reflect said update.

34. The method as recited in claim 31, further comprising:
detecting a change within a data processing system;
responsive to a determination that the change affects one of said set of first working definitions, modifying the one of said set of first working definitions to reflect the change; and

updating the runtime image of one of said plurality of applications defined by said one of said set of first working definitions.

35. The method as recited in claim 34, wherein the change is an event.

EVIDENCE APPENDIX

There is no evidence to be presented.

RELATED PROCEEDINGS APPENDIX

There are no related proceedings.